Spiral-Net with F1-based Optimization For Image-Based Crack Detection

Takumi Kobayashi

National Institute of Advanced Industrial Science and Technology, Japan
takumi.kobayashi@aist.go.jp

Abstract. Detecting cracks on concrete surface images is a key inspection for maintaining infrastructures such as bridge and tunnels. From the viewpoint of computer vision, the task of automatic crack detection poses two challenges. First, since the cracks are visually depicted by subtle patterns and also exhibit similar appearance to the other structural patterns, it is difficult to discriminatively characterize such less distinctive and finer defects. Second, the cracks are scarcely found, making the number of training samples for cracks significantly smaller than that of the other normal samples to be distinguished from the cracks. This is regarded as a class imbalance problem where the classifier is highly biased toward majority classes. In this study, we propose two methods to address these issues in the framework of deep learning for crack detection: a novel network, called Spiral-Net, and an effective optimization method to train the network. The proposed network is extended from U-Net to extract more detailed visual features, and the optimization method is formulated based on F1 score (F-measure) for properly learning the network even on the highly imbalanced training samples. The experimental results on crack detection demonstrate that the two proposed methods contribute to performance improvement individually and jointly.

1 Introduction

Crack detection on concrete surfaces is a primary task for inspecting infrastructures such as bridges and tunnels [24]. Since the degradation of those concrete structures is assessed by the length, width and density of the cracks [39], it is critical for the maintenance to finely record the situation of the cracks. As the number of concrete structures has been rapidly grown, an automatic crack detection attracts keen attention to reduce and replace the manual inspection, especially based on still images captured by digital cameras.

Cracks are rather related to lower-level image characteristics, being a bit apart from semantic objects of the targets in object detectors [28]. Thus, crack detection has been addressed mainly in the field of image processing by utilizing some heuristics of image derivative [8], wavelets [32] and morphological operations [38, 43]. The image processing technique, however, is not so enough to well distinguish the characteristics of cracks, producing lots of false positives, and therefore we demand the more discriminative approach for crack detection.

Discriminatively detecting cracks in an image is formulated into a pixel-wise binary classification task where each pixel is classified into the category either of *crack* or *non-crack* (normal). A naive approach toward the pixel-wise classification is a patch-based one as in most of neural network based crack detection methods [41, 4]. It predicts a class label at each pixel by classifying the features extracted from the patch and then processes whole image by means of sliding window. The method, however, has difficulty in detecting cracks minutely at pixel level, which is thus unsuitable for the purpose of finely depicting cracks.

The deep neural networks are successfully applied even to the pixel-wise classification. It is mainly addressed as semantic segmentation [20] by means of an encoder-decoder network [27, 35] which directly estimates a class label map of the same spatial dimensions as the input image. The encoder-decoder network whose shape resembles a hourglass has been further extended into U-Net [29] efficiently exploiting multi-resolution features via *skip connections* between the encoding and decoding layers. For such a semantic segmentation, fully convolutional network (FCN) [21] is also successfully applied with promising performance, leveraging discriminative object classification network [31].

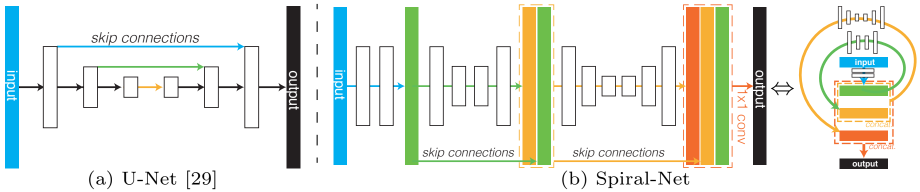
In the crack detection task of our focus, there are mainly two difficulties from the perspective of computer vision. 1) In contrast to semantic segmentation as well as object classification, where the targets exhibit distinctive image patterns, the crack detection on images has difficulty in charactering/describing the targets. The cracks are of less distinctive and finer patterns on image pixels, being vulnerable to confusion with the other structural patterns and superficial scratches which are irrelevant to degeneration of the (concrete) structures. 2) The task of detecting cracks also poses another challenge regarding class *imbalance* problem [9]. In the standard classification benchmark datasets, the distribution of training samples across classes are carefully designed so as to be close to uniform for facilitating classifier learning. On the other hand, the number of pixels belonging to cracks of the detection target is inherently too small compared with the other non-crack (normal) ones, which results in imbalanced training samples across two classes. Cracks are shown as *thin* lines and *rarely* found in healthy concrete images, causing the more highly imbalanced data than those used in the other semantic segmentation tasks and even edge detection [1, 36, 30]. The classifier trained on so imbalanced samples is biased toward the majority class while ignoring the characteristics of samples in the minor class.

In this paper, we propose two methods to address those two challenges naturally posed in the crack detection. For pixel-wise classification, we extend the U-Net [29] to extract detailed image characteristics in the encoder-decoder framework. The proposed network, called *Spiral-Net*, can produce a label map finely at pixel level by effectively exploiting diverse-level image features with keeping finer patterns to distinguish cracks from the others. And then the network is effectively learned on the imbalanced training samples where the crack pixels are significantly fewer than the non-crack pixels. While a standard loss such as binary cross-entropy usually employed in training networks suffers from the imbalanced-class samples, we propose an optimization approach based on

F1 score (F-measure) which has been mainly employed as an evaluation metric robust against the imbalanced classes. We derive from maximizing the F1 score an effective form of gradients for properly training a neural network over the imbalanced classes through back-propagation. These two proposed methods, Spiral-Net and the optimization approach, work individually and jointly in an end-to-end learning to improve performance of image-based crack detection.

1.1 Related Works

Network architecture. In the deep learning framework, the encoder-decoder network [27, 35] is successfully applied to such as semantic segmentation tasks. It is further extended to U-Net [29] by adding skip connections between the encoding and decoding layers to extract diverse-level image features of multi resolution. Both the encoding and decoding processes are improved in some works [10, 5, 35] to provide effective building blocks of the encoder-decoder networks. On the other hand, the overall network architecture is also improved beyond the simple encoder-decoder network. In [26], the encoder-decoder networks are sequentially stacked, being closely related to our method in terms of sequencing encoder-decoder networks (Sec. 2). It, however, differs in the two architectural points regarding skip connections and depths of the encoder-decoder networks. Those two characteristics in our network are useful for extracting finer image patterns of cracks. FCN [21] is also extended to cope with an edge detection task in [36] leveraging object classification network [31], and is empirically compared to our method on the crack detection task in the experiments (Sec. 4).
Imbalance problem. The methods to cope with the imbalanced classes are mainly categorized into two approaches of re-sampling and weighted loss.
Re-sampling. The imbalanced sample distribution can be corrected by either down-sampling samples in the majority class or over-sampling those of the minority class [6, 16, 22, 23, 42], which is sophisticated by [2] in the deep learning literature and is related to hard-negative mining [12, 7]. However, over-sampling can easily introduce undesirable noise and also have a risk of overfitting, while in down-sampling valuable information in training samples would be lost, which is a critical issue in training deep neural networks, a data-hungry procedure.
Weighted loss. On the other hand, there are methods to (re-)weight loss functions, called cost-sensitive approaches [9]. By assigning asymmetric weights on the losses across classes, one can remedy the high bias toward the majority class; that is, the losses for the majority classes are less-weighted, while those in the minor classes get larger weights to attract higher attention. They have been formulated for shallow learning methods, such as SVM [33], boosting [34] and random forest [17]. Such methods are recently investigated in the deep learning literature [3, 25, 37], though most of them follow the approaches applied to the shallow models; they take a relatively simple cost-sensitive approach based on re-balancing scheme using an inverse class frequency [36]. In [30], the target class is divided into sub-classes and the cross-entropy loss is regularized by using those sub-classes to cope with the imbalanced samples. In the method, however, the categorization into sub-classes is carefully designed and the number of sub-classes



(a) U-Net [29]

(b) Spiral-Net

Fig. 1. Network architecture of the proposed Spiral-Net in comparison to U-Net [29]. The U-Net (a) is unfolded in terms of its multiple skip connections into the Spiral-Net (b) where the encoder-decoder modules are stacked with increasing their depths and stacking their output feature maps, which results in a *spiral* shape; the stacked feature maps are finally classified into the class labels by 1 × 1 convolution.

is a hyper-parameter to be tuned by users. The recent method [19] focusing on dense object detection defines a focal loss to suppress the contributions of easily classified samples while shifting up the importance of hard samples, though introducing tunable hyper-parameters. Our approach to remedy the imbalance among classes is derived from F1 score through reformulating it toward differentiable loss from a probabilistic viewpoint [15], which thus exhibits clear difference from those previous works. It should be noted that our method is parameter-free and thereby easily embedded into the end-to-end learning.

2 Spiral-Net

We first describe our network architecture inspired from U-Net [29]. In recent years, image segmentation is often formulated in the framework of an encoder-decoder network [11] where an input image is encoded into effective features at the coarser resolutions and then decoded with increasing resolutions into the same spatial dimensions as the input but in the different domain such as labels. The simple encoder-decoder architecture is extended to U-Net [29] by introducing the skip connections between encoding and decoding layers, as shown in Fig. 1a, in order to leverage diverse image features of finer and coarser levels to predict the pixel-wise labels. In the U-Net (Fig. 1a), there are multiple paths from an input image to an output label map through skip connections with sharing convolution layers. For exploiting more detailed image characteristics, however, it may be necessary to apply respective encoding processes to extract features of various levels without sharing them. Therefore, we propose a network architecture, *Spiral-Net* (Fig. 1b), by unfolding the multiple paths in the U-Net.

The Spiral-Net is constructed by stacking multiple encoder-decoder subnetworks (modules) *sequentially*, which have been folded in the U-Net with sharing layers via skip connections. While the Spiral-Net is closely related to the stacked hourglass network [26] which is also a sequence of encoder-decoder modules, the proposed network has the following two characteristics.

First, it contains various encoder-decoder modules of diverse *depths* and they are sequentially aligned in an *increasing order* regarding their depths. In such an architecture, the first shallowest encoder-decoder is expected to work as rather simple image preprocessing, and then the more discriminative features of larger receptive fields are gradually extracted by the deeper encoder-decoder modules stacked in the latter positions. The former encoder-decoder module is of shallower depth, containing less parameters, so as to be effectively trained even though it is far from the loss layer, the source of gradients in back-propagation.

Second, as in DenseNet [13], we densely string skip connections between the feature maps produced by the encoder-decoder modules for directly exploiting features of diverse levels. It should be noted that the feature maps have the same spatial dimensions as in an input image, thus being fed into the successive encoder-decoder modules (Fig. 1b). Through concatenating the previous feature maps, the deeper encoder-decoder module receives the *wider* feature maps, which is favorable for extracting discriminative features. The input image is not propagated via the skip connection since the raw pixel values exhibit different characteristics from the other features. At the final classification layer, the 1×1 convolution is applied to predict pixel-wise class labels from the densely concatenated feature maps. Through these dense skip connections, the gradient information for updating parameters can be effectively back-propagated into the former encoder-decoder modules [13]. We can say that the proposed Spiral-Net is different from the DenseNet [13] in that the *encoder-decoder* module is embedded in each block, being also distinctive compared to FCN [21] which applies decoders (up-sampling) just as outgoing branches to output a map of class labels.

Based on these characteristics of the network, we can conceptually fold it into the *spiral* shape (Fig. 1b). In the Spiral-Net, the features of various depths can be extracted by the respective encoder-decoder modules unlike the U-Net which shares parts of the encoding processes, and our deeper encoder-decoder can effectively extract the features from the wider input feature map composed of diverse-level features. The Spiral-Net has flexibility in the encoder-decoder module so that we can choose various types of networks such as the ones based on dilated convolutions [40] and residual blocks [10, 35]; in this work, for computational efficiency, we employ the simple hourglass encoder-decoder which applies 3×3 convolutions to output a one-channel feature map, as shown in Table 1.

3 F1-based Optimization

The standard cross-entropy on imbalanced training samples biases the network toward the majority classes. To alleviate the class imbalance problem naturally found in the crack detection task, we propose an F1-based optimization method which is applicable to gradient-based optimization, i.e., back-propagation.

3.1 F1-based Loss

The F1 score (F-measure) is a standard metric to evaluate the classification performance in a robust manner against the imbalanced classes. Suppose two-class problem comparing *crack* (positive) with *non-crack* (negative) where the negative class is a majority. The F1 score is computed by

$$\mathtt{F1} = \frac{2\,\mathtt{prec} \cdot \mathtt{rec}}{\mathtt{prec} + \mathtt{rec}}, \tag{1}$$

where `prec` and `rec` indicate precision and recall rates based on the binary classification results, respectively. The `F1` depends on empirical counts of such as false positives, being obviously not differentiable, and thus has been applied mainly to evaluate the performance of the trained classifier. Toward a loss function, we first reformulate the definition (1) in a similar manner to [15]. Note, however, that our method is clearly different from [15] via the weighting scheme (Sec. 3.2).

At the i-th sample (pixel) assigned with the ground truth label $l_i \in \{-1, 1\}$, the posterior probability is computed by

$$\mathtt{p}(\hat{l}_i = 1) = \sigma(x_i) = \frac{1}{1 + \exp(-x_i)} \triangleq \sigma_i, \tag{2}$$

where $\hat{l}_i \in \{-1, 1\}$ indicates the predicted label by applying the sigmoid function σ to the feature x_i extracted at the i-th sample. Let N_1 and N_{-1} be the numbers of samples belonging to positive and negative classes, respectively, and the prior probabilities can be empirically estimated as

$$\mathtt{p}(l = 1) = \frac{N_1}{N_{-1} + N_1}, \quad \mathtt{p}(l = -1) = \frac{N_{-1}}{N_{-1} + N_1}. \tag{3}$$

Then, the precision and recall rates in the F1-score (1) are described as

$$\mathtt{rec} = \frac{\mathtt{p}(\hat{l} = 1, l = 1)}{\mathtt{p}(l = 1)} = \mathtt{p}(\hat{l} = 1 | l = 1) = \frac{1}{N_1} \sum_{j | l_j = 1} \sigma_j, \tag{4}$$

$$\mathtt{prec} = \frac{\mathtt{p}(\hat{l} = 1, l = 1)}{\mathtt{p}(\hat{l} = 1)} = \frac{\mathtt{p}(\hat{l} = 1 | l = 1) p(l = 1)}{\sum_{c \in \{-1,1\}} \mathtt{p}(\hat{l} = c | l = c) p(l = c)} = \frac{\sum_{j | l_j = 1} \sigma_j}{\sum_j \sigma_j}. \tag{5}$$

Therefore, the F1 score (1) is reformulated into

$$\mathtt{F1} = \frac{2 \sum_{j | l_j = 1} \sigma_j}{N_1 + \sum_j \sigma_j}, \tag{6}$$

where $0 \leq \mathtt{F1} \leq 1$. Beyond this naive F1 loss (6) which is also found in [15], we construct the loss by negative logarithm of the F1 score, $L = -\log(\mathtt{F1})$, of which the derivative is given by

$$\frac{\partial L}{\partial \sigma_i} = \begin{cases} -\left(\frac{1}{\sum_{j | l_j = 1} \sigma_j} - \frac{1}{N_1 + \sum_j \sigma_j} \right), & l_i = 1 \\ \frac{1}{N_1 + \sum_j \sigma_j}, & l_i = -1 \end{cases}, \tag{7}$$

where $\frac{1}{\sum_{j | l_j = 1} \sigma_j} - \frac{1}{N_1 + \sum_j \sigma_j} > 0$ due to that $N_1 > 0$ and $\sum_{j | l_j = 1} \sigma_j \leq \sum_j \sigma_j$.

3.2 F1-guided Gradient Weighting

It is possible to directly compute the gradient of the F1-based loss L w.r.t x_i by combining (7) and $\frac{\partial \sigma(x_i)}{\partial x_i} = \sigma(x_i)\{1 - \sigma(x_i)\}$ into the chain rule $\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x_i}$;

$$\frac{\partial L}{\partial x_i} = \begin{cases} -\left(\frac{1}{\sum_{j | l_j = 1} \sigma_j} - \frac{1}{N_1 + \sum_j \sigma_j} \right) \sigma_i (1 - \sigma_i), & l_i = 1 \\ \frac{1}{N_1 + \sum_j \sigma_j} \sigma_i (1 - \sigma_i), & l_i = -1 \end{cases}. \tag{8}$$

This straightforward approach, however, is not favorable for gradient-based optimization (back-propagation) since the gradient (8) contains $\sigma(x_i)\{1-\sigma(x_i)\}$ which unfavorably vanishes at the extreme predictions, $\sigma(x_i) \rightarrow \{1,0\}$; it is empirically shown in Sec. 4.4.

On the other hand, the most naive loss directly derived from the posterior probabilities is described by[1]

$$\tilde{L} = -\sum_{y\in\{-1,1\}}\sum_{i|l_i=y} \mathtt{p}(\hat{l}_i = y), \quad \frac{\partial \tilde{L}}{\partial \sigma_i} = \left\{ \begin{array}{rl} -1, & l_i = 1 \\ 1, & l_i = -1 \end{array} \right. . \tag{9}$$

We can regard the gradient (7) as the weighted version of (9) by introducing the weights derived from the F1 score (6), leading to the reformulation of $\frac{\partial L}{\partial x_i} = \left|\frac{\partial L}{\partial \sigma_i}\right| \frac{\partial \tilde{L}}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i}$. This point of view inspires us to apply the similar weighting approach to the commonly used cross-entropy loss of

$$\bar{L} = -\sum_{y\in\{-1,1\}}\sum_{i|l_i=y} \log(\mathtt{p}(\hat{l}_i = y)), \quad \frac{\partial \bar{L}}{\partial \sigma_i} = \left\{ \begin{array}{rl} -\frac{1}{\sigma_i}, & l_i = 1 \\ \frac{1}{1-\sigma_i}, & l_i = -1 \end{array} \right. . \tag{10}$$

Thus, we propose the following pseudo[2] gradients by weighting (10) with (7);

$$\mathtt{g}(x_i) = \left|\frac{\partial L}{\partial \sigma_i}\right| \frac{\partial \bar{L}}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i} = \begin{cases} -\left(\frac{1}{\sum_{j|l_j=1}\sigma_j} - \frac{1}{N_1+\sum_j \sigma_j}\right)(1-\sigma_i), & l_i = 1 \\ \qquad\qquad\frac{1}{N_1+\sum_j \sigma_j}\sigma_i \qquad\qquad\quad , & l_i = -1 \end{cases} . \tag{11}$$

In contrast to most of cost-sensitive methods [3, 25, 37], we directly impose weights on the gradients, not on the losses, though from the viewpoint of gradient-based optimization the cost-sensitive methods also produce weighted gradients through the weighted loss. In our end-to-end learning, the pseudo gradient (11) is back-propagated to update the parameters of the neural network.

The adaptive weights (7) work on the imbalance issue in a manner derived from optimizing the F1 score (6). The weight of the positive class is rewritten to

$$\left|\frac{\partial L}{\partial \sigma_i}\right|_{l_i=1} = \frac{1}{\sum_{j|l_j=1}\sigma_j} - \frac{1}{N_1+\sum_j \sigma_j} = \frac{N_1+\sum_{j|l_j=-1}\sigma_j}{\sum_{j|l_j=1}\sigma_j}\frac{1}{N_1+\sum_j \sigma_j}, \tag{12}$$

and its ratio to the weight $\left|\frac{\partial L}{\partial \sigma_i}\right|_{l_i=-1} = \frac{1}{N_1+\sum_j \sigma_j}$ of the negative class is given by

$$\mathtt{r} \triangleq \frac{\left|\frac{\partial L}{\partial \sigma_i}\right|_{l_i=1}}{\left|\frac{\partial L}{\partial \sigma_i}\right|_{l_i=-1}} = \frac{N_1+\sum_{j|l_j=-1}\sigma_j}{\sum_{j|l_j=1}\sigma_j} \geq 1, \tag{13}$$

[1] Actually, in the training, we divide the losses $\tilde{L}$ and $\bar{L}$ by the number of samples $N = N_1 + N_{-1}$, which is here omitted for simplicity.

[2] Unfortunately, there is no analytic loss function that produces the derivative (11); see the supplementary material.

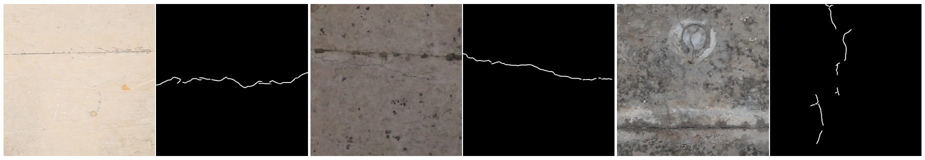


Fig. 2. Example images of cracks with label maps composed of binary values indicating crack pixels. They are of 512 × 512 pixels sampled from whole image of 5472 × 3678 pixels to focus on cracks. There are some structural patterns such as traces of concrete frame molds which are similar to but *not* cracks. Best viewed on the screen.

where the inequality comes from $\sum_{j|l_j=1}\sigma_j \leq N_1$. Thus, we can see that as is the case with the cost-sensitive methods, the minority samples belonging to the positive class are highly weighted while the negative ones are assigned with less weights. It is noteworthy that such weighting is theoretically induced from the probabilistic formulation of the F1 score (6), while a weighting scheme has been heuristically designed in the previous methods.

In particular, our weighing has the following properties which facilitate learning networks. In case that the classifier is biased toward the negative class (majority), resulting in $\sum_j \sigma_j \rightarrow 0$, the ratio $\mathtt{r}$ in (13) becomes larger to highly encourage the learning for the positive class. On the other hand, approaching to favorable classification of $\sum_{j|l_j=-1}\sigma_j \rightarrow 0$ and $\sum_{j|l_j=1}\sigma_j \rightarrow N_1$, the ratio $\mathtt{r}$ is close to 1, which realizes the equal weighting across positive and negative classes as is the case with the standard cross-entropy loss. This adaptive weighting scheme in the optimization enables the end-to-end learning to enjoy whole samples for effectively training networks unlike the re-sampling based methods.

In the case of mini-batch based optimization, we can consider the statistics on the mini-batch, and thereby all the ingredients in (11), $N_1, \sum_j \sigma_j$ and $\sum_{j|l_j=1}\sigma_j$, are computed over those samples within the mini-batch[3]. Since the proposed method (11) merely produces the weighted gradients, we can apply various types of effective optimization techniques used in the end-to-end learning.

4 Experimental Results

We evaluate the proposed methods of Spiral-Net (Sec. 2) and the F1-based optimization (Sec. 3.1) on a crack detection task; we assign a label either of *crack* or *non-crack* to every pixel in an image, which naturally induces the class imbalance problem while requiring finer image feature extraction to capture the visual characteristics of cracks.

4.1 Crack Dataset

We have collected still images at various locations such as tunnels, pillars and slabs of concrete bridges which are actually subject to inspection. The RGB-color

[3] In the preliminary experiment, we confirmed that the optimization using the globally cumulative statistics does not provide any performance improvement.

Table 1. Building blocks in U-Net and Spiral-Net. The encoders and decoders are implemented by convolution (`conv`) and transposed convolution (`convT`) of 3 × 3 filter without any padding nor cropping, respectively, which are followed by BatchNormalization and ReLU. We apply 2-pixel stride in `conv` and upsampling factor of 2 in `convT`.

<table>
<tr><th></th><th colspan="2">Encoder (conv)</th><th colspan="2">Decoder (convT)</th></tr>
<tr><th>Layer</th><th>Output dim.</th><th>Input size</th><th>Output size</th><th>Output dim.</th></tr>
<tr><td>1</td><td>64</td><td colspan="2">255×255</td><td>1</td></tr>
<tr><td>2</td><td>128</td><td colspan="2">127×127</td><td>64</td></tr>
<tr><td>3</td><td>256</td><td colspan="2">63×63</td><td>128</td></tr>
<tr><td>4</td><td>512</td><td colspan="2">31×31</td><td>256</td></tr>
<tr><td>5</td><td>512</td><td colspan="2">15×15</td><td>512</td></tr>
<tr><td>6</td><td>512</td><td colspan="2">7×7</td><td>512</td></tr>
<tr><td>7</td><td>512</td><td colspan="2">3×3</td><td>512</td></tr>
</table>

images of 5472 × 3678 pixels show the concrete surface containing a few cracks somewhere (Fig. 2), as well as the other objects, *e.g.*, pipes and steels, which are not eliminated for fairly evaluating the performance in the wild. It is noteworthy that the images are captured in the unconstrained situation, exhibiting high variations in terms of such as illumination and concrete colors. Then, the experts assigned the positive (*crack*) labels in a pixel-wise manner tracing the cracks by lines of roughly 3 pixel width, while the other pixels are regarded as belonging to the negative (*non-crack*) class. The crack pixels are scarce and the ratio of the numbers of crack and non-crack pixels is 1:450. In addition, as shown in Fig. 2, the crack patterns are not so distinct with exhibiting high similarity to the other structural patterns derived from such as molds and superficial scars. We used 278 images for training and 14 images for test which are picked up by crack inspectors, *not* expert of computer vision, to make fair evaluation of the performance on crack detection in real conditions; note that this dataset is as large as BSDS dataset [1] of edge detection in terms of number of pixels. The performance of crack detection is measured on each test image by average precision as well as precision, recall and F1 score which are computed based on the output (sigmoid function) of the network with the threshold of 0.5, and then we report those evaluation scores averaged across all the test images. Note that the these performance metrics are computed in a pixel-wise manner since the ground-truth label is assigned to each pixel.

4.2 Implementation Details

Inspired by the model used in pix2pix [14], we construct a vanilla U-Net which gradually downsizes an input image of 255 × 255 pixels by a factor of 2 as shown in Table 1; the 1st∼7th encoders and the 7th∼1st decoders are sequentially stacked with skip connections (Fig. 1a). The Spiral-Net is also composed of the same building blocks as in Table 1; the encoder-decoder module of d depth is built by sequentially stacking the 1st∼d-th encoders and the d-th∼1st decoders.

In training, we randomly pick up 32 image patches of 255 × 255 pixels from training images to shape the mini-batch with random flipping either horizontally or vertically. At each epoch, such sampling is repeated 256 times *per image* so as to roughly cover the whole image of 5472 × 3678 pixels by using the patches;

Table 2. Performance results (%) by Spiral-Net of various depth orders. The numbers in the second column indicate the depths of the stacked encoder-decoder modules.

Architecture	Order of depths	mAP	F1	Precision	Recall
Increasing ╱	1-2-3-4-5-6-7	**80.91**	**71.76**	86.56	62.89
Decreasing ╲	7-6-5-4-3-2-1	0	0	0	0
Triangle ⋀	1-2-3-4-3-2-1	78.92	69.35	87.33	59.23
	1-3-5-7-6-4-2	79.03	70.82	85.39	62.38
Uniform —	1-1-1-1-1-1-1	53.67	43.85	82.16	32.06
	2-2-2-2-2-2-2	77.56	67.51	87.29	57.05
	3-3-3-3-3-3-3	78.33	67.99	88.71	58.50
	4-4-4-4-4-4-4	74.77	66.96	81.34	58.79
	5-5-5-5-5-5-5	72.08	63.77	84.22	55.02
	6-6-6-6-6-6-6	0	0	0	0
	7-7-7-7-7-7-7	0	0	0	0

Table 3. Performance results on Spiral-Net with various types of skip connections.

Connection	mAP	F1	Precision	Recall
Concatenation	**80.91**	**71.76**	86.56	62.89
Sum	79.16	71.14	87.12	61.46
None	0	0	0	0

thereby, we receive $2224 = 256 \cdot 278/32$ mini-batches per epoch in the training. The network is trained by applying Adam optimizer [18] with the learning rate of 0.0001 and momentum of 0.9 over 200 epochs.

4.3 Performance Analysis on Spiral-Net

We evaluate the Spiral-Net (Sec. 2) in terms of the network configuration, by training all the networks based on the standard binary cross-entropy loss.

The Spiral-Net stacks the encoder-decoder modules of different depths (Fig. 1b). Thus, the network architecture is controlled by the sequential order of those modules; the depths of the stacked encoder-decoder can be designed as follows.

– As described in Sec. 2, the modules are sequentially aligned so that their depths are in an increasing order.

– In contrast, it is also possible to stack them in a decreasing order.

– An intermediate design between those two could be the one in which the depths are first increasing and then decreasing like a triangle shape.

– On the other hand, the simplest architecture is that all the modules have the uniform (identical) depth.

The performance results are shown in Table 2. We can see that the performance is significantly degraded by locating the deeper encoder-decoder module early in the network (decreasing and deeper uniform). As discussed in Sec. 2, such deeper module can not be properly trained since it is far from the loss layer and receives *narrower* feature maps. Actually, the networks of decreasing and uniform with depths of 6 and 7 are improperly learned to always output labels of negative class which is the majority in the dataset; in those cases, the performances are shown as all 0's. In the uniform architecture, the moderately deep encoder-decoders work well while the shallower and deeper ones provide poor performance. On the other hand, the networks gradually increasing depths

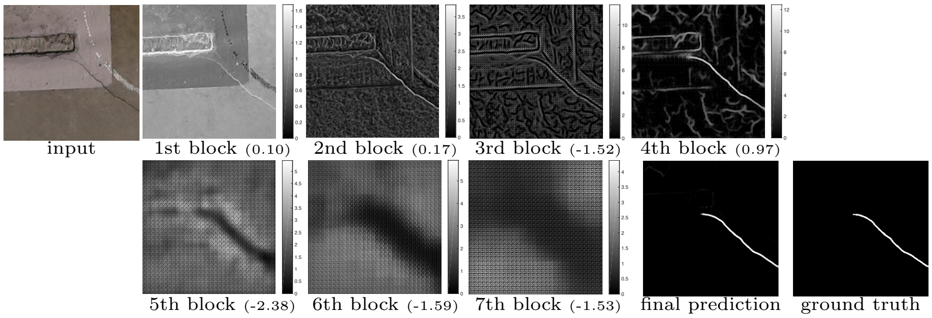


Fig. 3. Intermediate feature maps by the 1st∼7th encoder-decoder modules. The number in parentheses indicates the aggregation weight at the final 1×1 convolution layer.

in increasing and triangle produce the better performance than the uniform one, and especially, the best performance is achieved by the increasing order.

Next, we evaluate the following types of skip connection on the Spiral-Net of the increasing depth order.

– The feature maps propagated via skip connections are concatenated along the channel dimension to increase the number of channels in a way of DenseNet [13].

– The propagated feature maps are summed up as in ResNet [10] with keeping the number of channels. Note that any encoder-decoder modules output one-channel feature maps (see Table 1).

– We do not apply any skip connections in the network in a similar way to [26].

The performance comparison is shown in Table 3. Without any skip connections (none), in this imbalanced data, the network is not properly learned due to that the gradient information is not effectively back-propagated. The skip connections based on sum and concatenation remedy the issue, producing favorable performance. In comparison with sum-based connection, the concatenation one provides the *wider* feature maps which contribute to further performance improvement, as described in Sec. 2.

These experimental results quantitatively validate the proposed Spiral-Net that stacks encoder-decoder modules of increasing depths with providing wider feature maps due to concatenation-based skip-connections.

We then qualitatively analyze how the Spiral-Net detects cracks by showing the intermediate one-channel feature maps together with the last 1×1 convolution weights to merge them; the respective encoder-decoder modules produce the non-negative one-channel maps as shown in Fig. 3. The 1st&2nd modules work as lower-level image processing like pixel-value enhancement and derivative computation; these outputs less contribute to the final prediction due to their smaller weights. Based on those low-level features, the 3rd&4th modules detect crack-like structures, extracting candidates for cracks with the positive weight at the 4th module while suppressing the other regions by the negative weight at the 3rd one. Finally, the 5th∼7th modules detect cracks rather semantically and effectively eliminate the false positives by assigning large negative weights on the non-crack regions. These sequential processes are quite reasonable by integrating lower-level image processing and higher-level classification in the Spiral-Net.

4.4 Comparison to Other Methods

We compare the proposed methods, Spiral-Net and the F1-based optimization (11), to the other methods in terms of a network and a loss function. In the training, the F1-based optimization is applied by simply replacing the gradients of the cross-entropy loss with the pseudo weighted gradients (11), and thus is applicable to any types of networks including the proposed Spiral-Net. The optimization method is formulated so as to cope with the class imbalance problem, which is particularly found in this crack detection task; note again that the number of crack pixels is far smaller ($\approx 1/450$) than that of non-crack pixels. Table 4 shows the performance results of various networks trained on various losses.

As to networks, we compare the Spiral-Net to the U-Net [29] and HED [36] all of which are trained on the crack dataset. HED is proposed based on FCN [21] for detecting (semantic) *edges* whose shapes are formed as thin lines similarly to cracks. In the work [36] which tackles edge detection tasks, HED is fine-tuned from the VGG pre-trained model [31] based on the cross-entropy loss weighted by the inverse of class frequency. However, we can see that, on any types of losses, so fine-tuned HED is inferior to the one trained from scratch on this crack dataset. While the semantic edge detection is closely related to object recognition on which the VGG pre-trained model works, the crack detection is not so dependent on the object recognition but is rather formulated as lower-level image processing taking into account the finer image structure, though both tasks aim to produce thin lines, *edge* and *crack*. Thus, although the HED fine-tuned from the image classification model (VGGnet) is suitable for edge detection tasks, it largely degrades performance on this crack detection task. On the other hand, while the U-net is comparable to the HED trained from scratch, the proposed Spiral-Net outperforms those on diverse types of losses, demonstrating that the network effectively extracts detailed image characteristics of crack patterns.

Next, the F1-based optimization method is compared to the other types of loss functions: the widely used cross-entropy loss $\bar{L}$ in (10), the one weighted by inverse of class frequency, and the focal loss [19]. The latter two losses are developed from the cross-entropy loss via weighting; the weights by inverse class frequency are introduced as $\check{L} = -\sum_{y\in\{-1,1\}} \frac{1}{N_y} \sum_{i|l_i=y} \log(\mathtt{p}(\hat{l}_i = y))$, and the simple weighting scheme is recently more sophisticated in the focal loss [19] as $\acute{L} = -\sum_{y\in\{-1,1\}} \alpha_y \sum_{i|l_i=y} \{1 - \mathtt{p}(\hat{l}_i = y)\}^\gamma \log(\mathtt{p}(\hat{l}_i = y))$ where $\alpha_1 = \alpha$, $\alpha_{-1} = 1 - \alpha$, and γ, α are the parameters to be determined by users; we set $\gamma = 2$ and $\alpha = 0.25$ as suggested in [19] and then tuned it to $\alpha = 0.5$. These losses are applied to the above-mentioned networks, and the performance results are shown in Table 4.

The cross-entropy loss makes the detector focus on the majority class of *non-crack* pixels, which results in relatively high precision and low recall as shown in Table 4a. On the contrary, through weighting by the inverse class frequency, the detector is highly biased to the minor class of *crack* pixels, producing high recall and low precision (Table 4b); it shows the difficulty in manually tuning the class weight in this highly imbalanced data. And, even the focal loss [19] degrades performance compared to the cross-entropy loss (Table 4cd). Note that

Table 4. Performance comparison in terms of networks and loss functions.

	(a) Cross-entropy $\bar{L}$				(b) Cross-entropy weighted by inverse class frequency $\check{L}$			
network	mAP	F1	Precision	Recall	mAP	F1	Precision	Recall
Spiral-Net (ours)	80.91	71.76	86.56	62.89	80.11	29.19	17.48	97.99
U-net [29]	73.12	64.78	78.69	55.63	73.02	21.80	13.83	90.83
HED(fine-tune) [36]	63.47	18.83	90.83	11.16	67.44	22.51	12.98	94.49
HED(scratch) [36]	72.47	61.70	85.85	51.61	72.39	28.87	18.66	92.29

	(c) Focal Loss $\acute{L}$ [19] ($\alpha = 0.2$)				(d) Focal Loss $\acute{L}$ [19] ($\alpha = 0.5$)			
network	mAP	F1	Precision	Recall	mAP	F1	Precision	Recall
Spiral-Net (ours)	68.01	44.20	91.24	31.08	75.77	65.17	86.87	54.46
U-net [29]	63.40	41.45	90.46	28.98	64.99	55.54	82.10	45.84
HED(fine-tune) [36]	53.86	0.98	64.21	0.50	56.34	12.37	90.94	7.01
HED(scratch) [36]	69.99	31.09	95.25	19.77	71.22	57.18	85.68	45.98

	(e) F1 in (8)				(f) Pseudo-F1 (ours) in (11)			
network	mAP	F1	Precision	Recall	mAP	F1	Precision	Recall
Spiral-Net (ours)	68.81	74.92	81.90	69.95	**85.61**	**79.04**	78.44	79.81
U-net [29]	63.06	68.97	81.63	64.85	77.16	69.48	77.67	68.34
HED(fine-tune) [36]	0	0	0	0	69.02	62.32	66.05	61.14
HED(scratch) [36]	0	0	0	0	76.96	69.00	70.43	71.20

Table 5. Parameter sizes of networks. The *wide* U-Net is constructed by increasing the number of channels in the U-Net.

Network	HED	U-Net	Spiral-Net	*wide* U-Net
# of parameter	14.1M	22.4M	39.7M	39.9M
mAP	76.96	77.16	85.61	77.01
F1	69.00	69.48	79.04	69.69

the focal loss indirectly corrects the class imbalance through suppressing the effect of *easy* negative samples which would occupy most of training samples causing the class imbalance. Such assumption holds on the object detection task addressed in [19] where the target objects exhibit clear difference in their visual appearance compared with most of background samples. In the crack detection, however, the target cracks are less distinctive in comparison with the other image patterns, reducing the number of *easy* samples, which would be the main reason why the focal loss is inferior even to the cross-entropy loss. In contrast, the proposed method favorably improves the performance of all the networks, including Spiral-Net (Table 4f), while the straightforward F1-based loss deteriorates performance (Table 4e). As discussed in Sec. 3.2, the gradients (8) of the F1-based loss contains the term of $\sigma_i(1 - \sigma_i)$ which unfavorably hampers learning; especially, the HEDs are improperly learned since it is trapped in the state extremely biased toward negative class (Table 4e). The proposed method adaptively tune the weights for the gradients based on the optimization of F1 score while avoiding the unfavorable formulation in the gradients to effectively improve performance (Table 4f); it is also balanced in terms of precision/recall.

The Spiral-Net trained by the F1-based optimization method achieves 85.61% which significantly outperforms 73.12% of the baseline method of U-Net trained by the cross-entropy loss. It is noteworthy that the performance improvement by the Spiral-Net comes from the architecture itself, not the increased size of parameters, as shown in Table 5. The examples of the detected cracks are shown in Fig. 4, demonstrating that even less distinctive cracks can be detected while being insensitive to the other patterns similar to cracks.

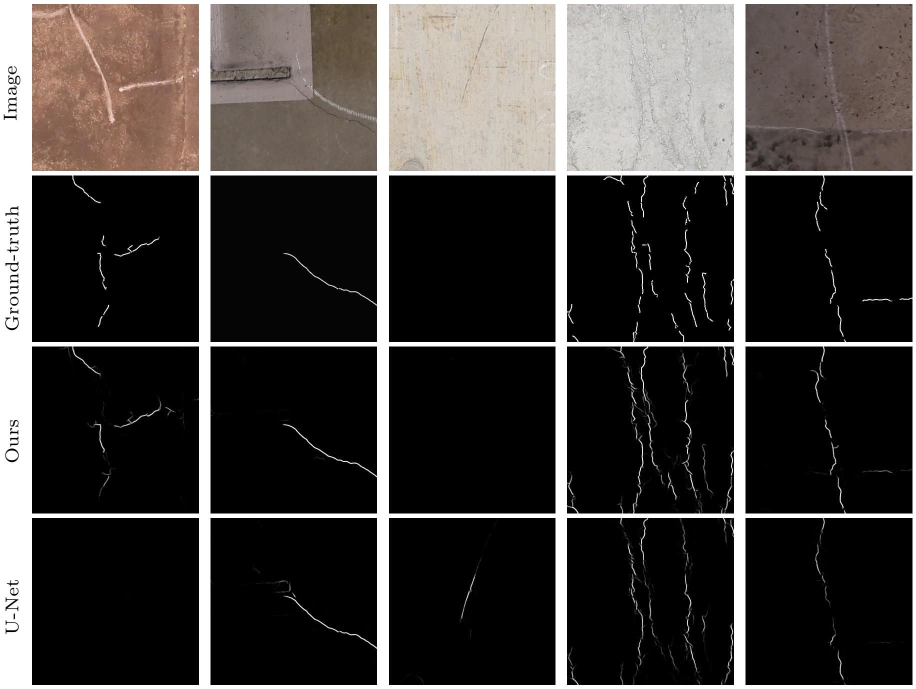


Fig. 4. Examples of detected cracks on the test images, showing the sigmoid output $[0, 1]$ in gray scale. The baseline method of U-Net trained by cross-entropy loss failed to detect some cracks while producing the false positives. In contrast, the proposed method of Spiral-Net trained by F1-based optimization favorably detects cracks exhibiting well correspondence with the ground truth. Best viewed on the screen.

5 Conclusion

We have proposed the Spiral-Net and the optimization method based on F1 score for detecting cracks in an image. The cracks are of finer patterns and can be scarcely found on concrete surfaces, posing a class imbalance problem. The Spiral-Net is constructed by sequentially stacking encoder-decoder modules of increasing depths with skip connections for feature maps in order to extract detailed image features of cracks. In learning the network on the highly imbalanced training samples, we adaptively weight the gradients of the cross-entropy loss and the weights are theoretically derived from optimizing F1 score which is robust against the imbalanced classes. The experimental results on image-based crack detection demonstrate the effectiveness of the two proposed methods, respectively, as well as their joint contribution to performance improvement.

Acknowledgment: The author thanks Takeshi Nagami, Hisashi Sato and Yohei Hayasaka for their great effort to build the crack dataset. This work is based on a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI **33**(5), 898–916 (2011)
2. Bulo, S.R., Neuhold, G., Kontschieder, P.: Loss max-pooling for semantic image segmentation. In: CVPR. pp. 7082–7091 (2017)
3. Caesar, H., Uijlings, J.R.R., Ferrari, V.: Joint calibration for semantic segmentation. In: BMVC (2015)
4. Cha, Y.J., Choi, W., Büyüköztürk, O.: Deep learning-based cracking damage detection using cnns. Computer-Aided Civil and Infrastructure Engineering **32**(5), 361–378 (2017)
5. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)
7. Dong, Q., Gong, S., Zhu, X.: Class rectification hard mining for imbalanced deep learning. In: ICCV. pp. 1869–1878 (2017)
8. Fujita, Y., Hamamoto, Y.: A robust automatic crack detection method from noisy concrete surfaces. Machine Vision and Applications **22**, 245–254 (2011)
9. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
12. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR. pp. 5375–5384 (2016)
13. Huang, G., Liu, Z., Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 5967–5976 (2017)
15. Jansche, M.: Maximum expected f-measure training of logistic regression models. In: HLT. pp. 692–699 (2005)
16. Jeatrakul, P., Wong, K.W., Fung, C.C.: Classification of imbalanced data by combining the complementary neural network and smote algorithm. In: International Conference on Neural Information Processing. pp. 152–159 (2010)
17. Khoshgoftaar, T.M., Golawala, M., Hulse, J.V.: An empirical study of learning from imbalanced data using random forest. In: ICTAI. pp. 310–317 (2007)
18. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
19. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: ICCV. pp. 2999–3007 (2017)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. pp. 3431–3440 (2015)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. pp. 3431–3440 (2015)
22. Maciejewski, T., Stefanowski, J.: Local neighborhood extension of smote for mining imbalanced data. In: ICDM. pp. 104–111 (2011)
23. Mani, I., Zhang, I.: Knn approach to unbalanced data distributions: a case study involving information extraction. In: Workshop on learning from imbalanced datasets (2003)

24. Mohan, A., Poobal, S.: Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal (2017). https://doi.org/https://doi.org/10.1016/j.aej.2017.01.020
25. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: Feed-forward semantic segmentation with zoom-out features. In: CVPR. pp. 3376–3385 (2015)
26. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV. pp. 483–499 (2016)
27. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV. pp. 1520–1528 (2015)
28. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
30. Shen, W., Wang, X., Wang, Y., Bai, X., Zhang, Z.: Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In: CVPR. pp. 3982–3991 (2015)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
32. Taha, M.M.R., Noureldin, A., Lucero, J.L., Baca, T.J.: Wavelet transform for structural health monitoring: A compendium of uses and features. Structural Health Monitoring (5), 267—-295 (2006)
33. Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S.: Svms modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics **39**(1), 281–288 (2009)
34. Ting, K.M.: A comparative study of cost-sensitive boosting algorithms. In: ICML. pp. 983–990 (2000)
35. Wojna, Z., Ferrari, V., Guadarrama, S., Silberman, N., Chen, L.C., Fathi, A., Uijlings, J.: The devil is in the decoder. In: BMVC (2017)
36. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV. pp. 1395—1403 (2015)
37. Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: CVPR. pp. 3781–3790 (2015)
38. Yamaguchi, T., Hashimoto, S.: Fast crack detection method for large-size concrete surface images using percolation-based image processing. Machine Vision and Applications **21**, 797—809 (2010)
39. Yang, Y.S., Yang, C.M., Huang, C.W.: Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis. Advances in Engineering Software **83**, 99–108 (2015)
40. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
41. Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J.: Road crack detection using deep convolution neural network. In: ICIP. pp. 2791–2799 (2016)
42. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering **18**(1), 63–77 (2006)
43. Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S.: Cracktree: Automatic crack detection from pavement images. Pattern Recognition Letters **33**(3), 227—238 (2012)